Java Program 1

Concepts

- Using packages
- Command line arguments
- Extracting an `int` from a String
- `for` loop
- `DecimalFormat`
- Tab character (“`\t`”)
- `Math` class

Description

This program will generate a short table of a few trigonometric functions (see Program Output on page 2) using static methods in the `java.lang.Math` class. The number of rows to be printed is read in from the command line and a “Usage” statement is printed if a number is not entered (check the length of the `args` array for this).

Each row will consist of the values *x*, *sin(x)*, *cos(x)*, and *tan(x)*. The *x* values start with 0 and have increments of $2\pi/n$ where *n* is the number entered on the command line.

Write a class called `TrigTable` in a package called `oa3302`. That means your program must be in a file called `TrigTable.java` in a subdirectory called `oa3302`. To compile from an MS-DOS prompt, change to the directory just above oa3302 and give the relative path name to the compiler (i.e. `javac oa3302\TrigTable.java`). To execute, stay in the same directory above oa3302 and give the fully-qualified class name to the interpreter (i.e. `java oa3302.TrigTable`). If your are using an integrated development environment (IDE) such as JBuilder, then follow the instructions for using that software to create and execute the program.

To extract an `int` from a `String`, combine the following two facts. There is a static method in `Integer` called `valueOf(String)` that returns an `Integer` object corresponding to the `String` passed to it; there is an instance method of `Integer` called `intValue()` that returns an `int` corresponding to the value for that object.

To make the columns line up, use the tab (“`\t`”) character and `DecimalFormat`. `DecimalFormat` is instantiated with a `String` indicating how you want to display the number. In this case the `String` should be `“ 0.000;-0.000”` (note the space). Read the JavaDoc for `DecimalFormat` for details on how it works. You should probably do the formatting last.

On the output, note the “Usage” statement. If your program expects input it does not receive, then you should be as “user-friendly” as possible, especially if you are the primary user of the program. In this case, use the length of the `args` array to see if the user has entered the correct number of command line parameters (1 in this case):[1]

```
if (args.length < 1) {
      System.err.println(“Usage: java oa3302.TrigTable <Number Rows>”);
      return;
}
```

1. Note that returning from main effectively exits the program.

Sample Program Output

```
> java oa3302.TrigTable
Usage: java TrigTable <Number Rows>
> java oa3302.TrigTable 15
  x     Sin(x)  Cos(x)  Tan(x)
  -     ------  ------  ------
 0.000   0.000   1.000   0.000
 0.419   0.407   0.914   0.445
 0.838   0.743   0.669   1.111
 1.257   0.951   0.309   3.078
 1.676   0.995  -0.105  -9.514
 2.094   0.866  -0.500  -1.732
 2.513   0.588  -0.809  -0.727
 2.932   0.208  -0.978  -0.213
 3.351  -0.208  -0.978   0.213
 3.770  -0.588  -0.809   0.727
 4.189  -0.866  -0.500   1.732
 4.608  -0.995  -0.105   9.514
 5.027  -0.951   0.309  -3.078
 5.445  -0.743   0.669  -1.111
 5.864  -0.407   0.914  -0.445
```

Deliverables

Turn in hard copies of your source code (meeting the OA3302 programming standards) and the output from your program using 15 on the command line, as above. *Note*: To print the output, redirect it to a file and print that file; that is, print the file `TrigTable.out` produced by the following:

```
java oa3302.TrigTable 30 > TrigTable.out
```